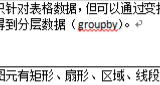
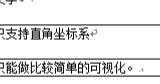
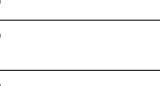

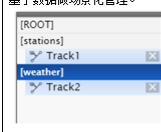
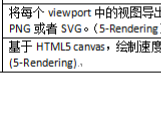



胡万祺

一、本周工作

- 1、写需求文档和分析文档。
- 2、重新阅读 lyra 和 ivisdesigner 两篇文章，总结两个工具的优点和缺点。

	VizDesigner	Lyra
缺点	不能设计自适应的图元。例如 bar chart 里面的每个 bar 是需要根据视图的高度（或宽度）除以 bar 条目的数量得出的。（4.3）	只针对表格数据，但可以通过变换得到分层数据（groupby）。 
	不能处理图元间的依赖关系，每个图元只能单独地去做数据映射。（4.3）	图元有矩形、扇形、区域、线段、文字。 
	做不了递归布局的图（例如 treemap）。（4.3）	只支持直角坐标系 
	不支持径向布局（4.3）	只能做比较简单的可视化。
	坐标轴没有属性编辑功能。（4.3）	
	不能指定数据点的顺序，文章中提出可以用某个维度去排序解决这个问题。（4.3）	
	对各种 paths 的定义不全面：按照特定顺序连接各个数据点；还有 path 和 path-之间区域填充等。（line、LineThrough）	
	学习成本太高（@），文章说是因为没有维护在线文档；个人认为还是因为系统的每个板块的使用都没有提示。	
	数据的绑定用下拉框的形式，分散到很多个模块，比如选定工具的地方、编辑图元属性的地方。	

优点	(贡献) 能处理不同类型的数据，交互式地设计、绑定、组合和关联不同的可视化图元。 数据集用 JSON 对象表达，主要处理 csv 表格数据，也能处理分层数据。每个数据项有提示数据类型、数据量。(5-Input Format) 	(贡献) 数据处理 pipeline：通过拖拽等交互进行图元编辑和数据绑定。 有很系统性的数据 pipeline，数据选择、变换、过滤都放在一起。 
	基于数据集场景化管理。  将每个 viewport 中的视图导出为 PNG 或者 SVG。(5-Rending) 基于 HTML5 canvas，绘制速度快。(5-Rending)	分层管理图元。  Vega 语言表达式可视化组成。

- 3、用 Xmind 列出 visketch 整体架构。

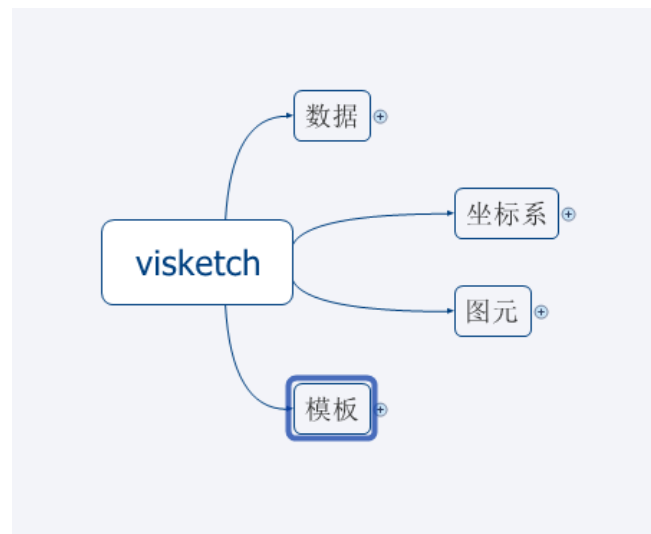


图 1: visketch 整体架构

其中数据部分可细分为：

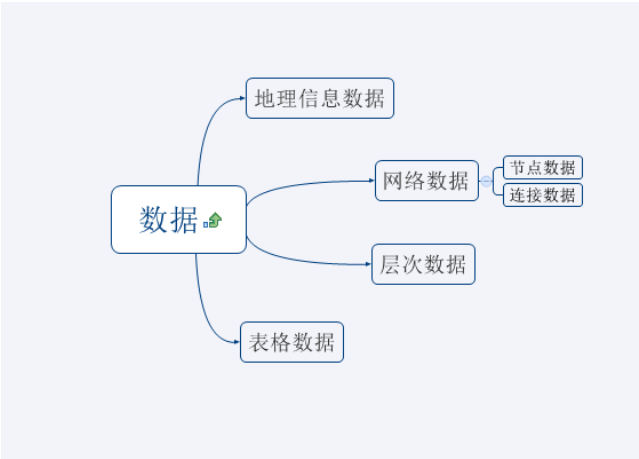


图 2：数据分类

坐标轴部分可细分为：

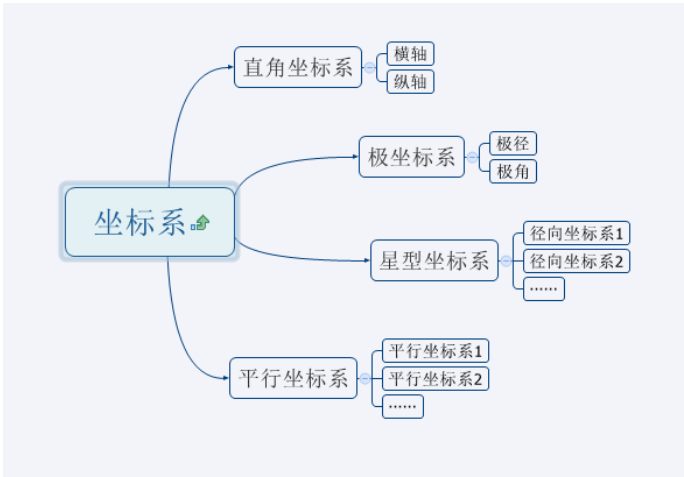


图 3：坐标轴分类

图元部分可细分为图 4 所示。圆形表示可以绑定数据的属性接口。**红色字体的属性接口是决定该图元的绘制个数的（视图的位置属性是否决定绘制个数这个可能还要另作讨论）。**有绿色旗子的属性为开放的可编程接口。

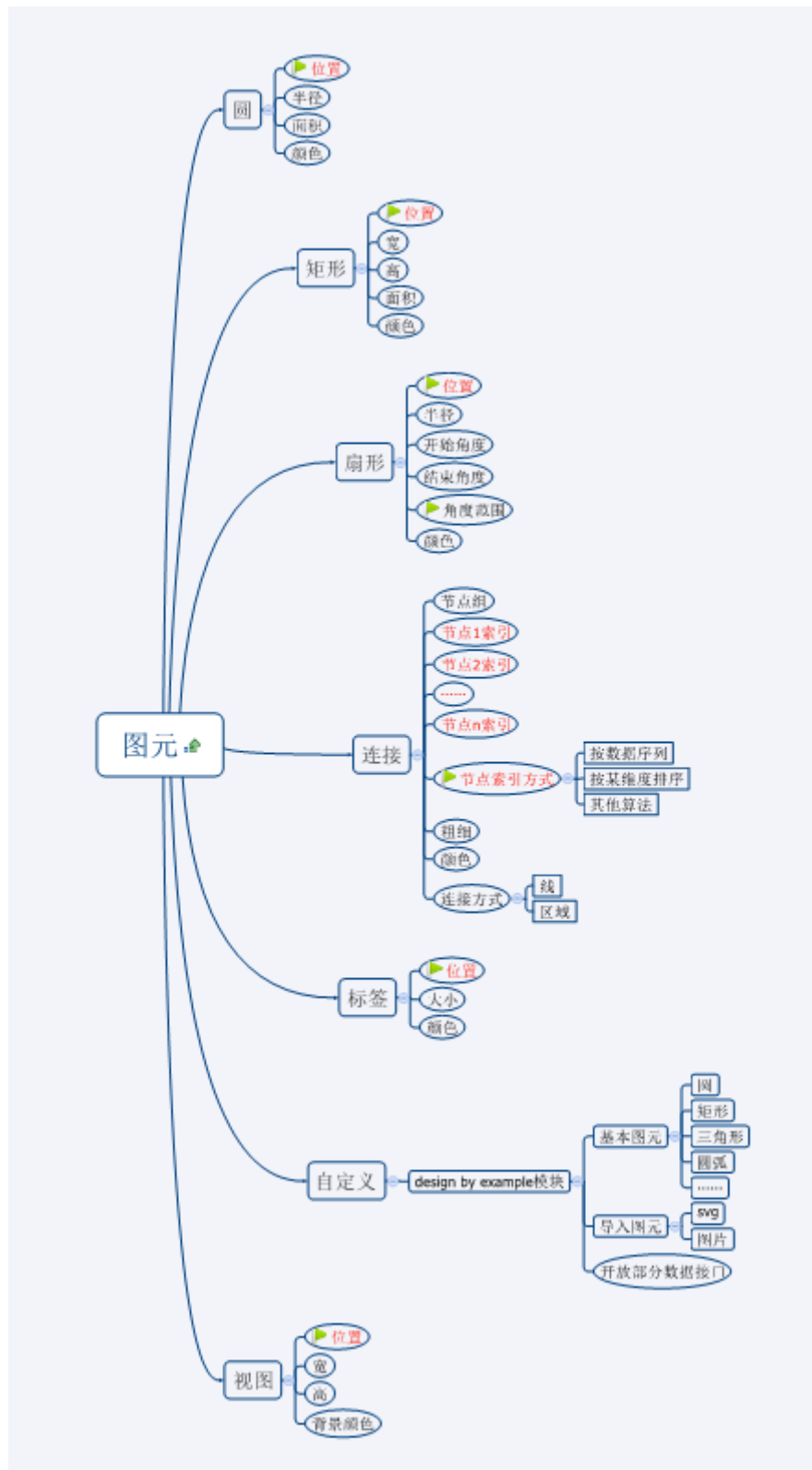


图 4：图元分类

系统还提供模板，图 5 中圆形为用户可绑定的属性接口，蓝色填充的圆形为自动绑定的属性接口。**平行坐标图处分析还有待讨论。**

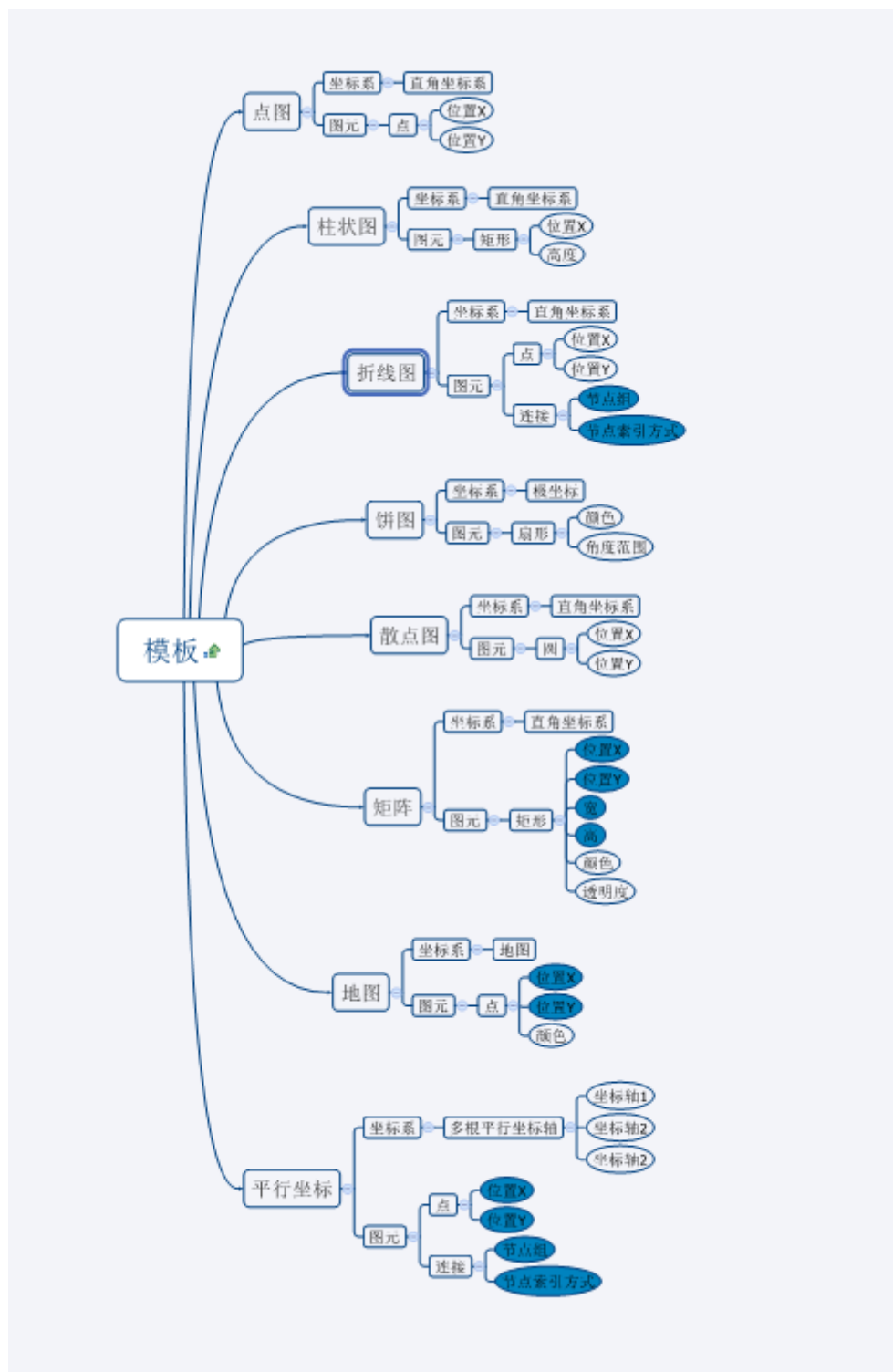


图 5：模板分类

二、下周工作

- 1、组织组会。
- 2、做好系统实现的分工。